Ad Hoc Networks xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Trust threshold based public key management in mobile ad hoc networks

Q1 Jin-Hee Cho[a,*], Ing-Ray Chen[b], Kevin S. Chan[a]

[a] U.S. Army Research Laboratory, Adelphi, MD 20783, USA
[b] Department of Computer Science, Virginia Tech, Falls Church, VA 22043, USA

ARTICLE INFO

Article history:
Received 5 October 2015
Revised 14 January 2016
Accepted 21 February 2016
Available online xxx

Q2 *Keywords:*
Public key management
Mobile ad hoc networks
Trust
Risk
Private key
Public key
Certificate authority

ABSTRACT

Public key management in mobile ad hoc networks (MANETs) has been studied for several decades. However, the unique characteristics of MANETs have imposed great challenges in designing a fully distributed public key management protocol under resource-constrained MANET environments. These challenges include no centralized trusted entities, resource constraints, and high security vulnerabilities. This work proposes a fully distributed trust-based public key management approach for MANETs using a soft security mechanism based on the concept of trust. Instead of using hard security approaches, as in traditional security techniques, to eliminate security vulnerabilities, our work aims to maximize performance by relaxing security requirements based on the perceived trust. We propose a composite trust-based public key management (CTPKM) with the goal of maximizing performance while mitigating security vulnerability. Each node employs a trust threshold to determine whether or not to trust another node. Our simulation results show that an optimal trust threshold exists to best balance and meet the conflicting goals between performance and security, by exploiting the inherent tradeoff between trust and risk. The results also show that CTPKM minimizes risk (i.e., information leakout) using an optimal trust threshold while maximizing service availability with acceptable communication overhead incurred by trust and key management operations. We demonstrate that CTPKM outperforms both existing non-trust-based and trust-based counterparts.

Published by Elsevier B.V.

1. Introduction

In resource-constrained mobile ad-hoc networks (MANETs), it is inefficient to employ cryptographic techniques for key management due to high computation and communication overhead as well as network dynamics that could require frequent key reassignments. In addition, the unique nature of MANETs does not allow any centralized trusted certificate authority (CA) to deal with all key management operations, including key generation, distribution, update, and revocation. Essentially, it is infeasible to build a system using hard security approaches (e.g., encryption or authentication techniques) to meet the dual goals of performance (i.e., efficiency) and security due to the inherent tradeoff. In this work, we take a soft security approach by applying the concept of trust to meet both performance and security requirements.

The concept of "trust" originally is derived from social science and defined as the degree of a subjective belief about the behaviors of a particular entity [1]. Blaze et al. [2] first introduced the term "trust management" and identified it as a separate component of security services in networks. They explained that "Trust management provides a unified approach for specifying and interpreting security policies, credentials, and relationships." Trust

* Corresponding author. Tel.: +1 571 232 3817.
E-mail addresses: jinhee.cho@us.army.mil, jinheechogwb@yahoo.com, jinheechogwb@gmail.com (J.-H. Cho), irchen@vt.edu (I.-R. Chen), kevin.s.chan@us.army.mil (K.S. Chan).

http://dx.doi.org/10.1016/j.adhoc.2016.02.014
1570-8705/Published by Elsevier B.V.

management in MANETs is needed when participating nodes, without any previous interactions, desire to establish a network with an acceptable level of trust relationships among themselves.

Trust management, including trust establishment, trust update, and trust revocation, in MANETs is more challenging than in traditional centralized environments [3]. First of all, collecting trust evidence to evaluate trustworthiness is difficult due to topology changes caused by node mobility/failure. Further, resource constraints often confine trust assessment process only to local information. The dynamic nature and characteristics of MANETs result in uncertain, incomplete trust evidence, which is continuously changing over time [3,4]. Cho et al. [5] comprehensively surveyed trust management in MANETs recognizing that trust originates from various domains including psychology, sociology, economics, philosophy, organizational theory, and so on. Cho et al. [5] suggested that the following properties be considered when designing trust-based MANET protocols: (1) potential risk; (2) context-dependency; (3) each party's own interest (e.g., utility/payoff based on rational selfishness); (4) learning based on cognition/experience; and (5) system reliability.

Trust management has diverse applicability in many decision making situations including intrusion detection [6,7], authentication, access control, key management, isolating misbehaving nodes for effective routing [6,8,9], and many other purposes [9]. In addition, the concept of multidimensional trust recently has been explored in networking and computing research areas and applied in various security services [6,7,10–13]. Bao et al. [6,7] proposed trust-based secure routing and intrusion detection mechanisms for wireless sensor networks by considering multiple dimensions of trust. Cho et al. [10,11] and Chen et al. [12] proposed trust management protocols for MANETs or delay tolerant networks considering multiple trust components. However, the above works [6,7,10–12] did not consider trust-based public key management while assuming a pre-loaded private/public key pair in each node. Very recently Mahmoud et al. [13] proposed trust-based secure and reliable routing for heterogeneous multihop wireless networks where competence and reliability of a node are estimated and used to derive the node trust level that can be used in routing decisions. However, Mahmoud et al. [13] assume the existence of a centralized offline trusted party to deal with public key management including issuance, distribution, and update of a public/private key pair to nodes in the network. Our work uses distributed peer-to-peer trust evaluation for public key management using three trust dimensions capturing the unique aspects of trust in a MANET.

In this paper, we propose a composite trust-based distributed key management algorithm (CTPKM) for MANETs without using a centralized trusted CA. Our approach falls under the category of certificate-based public key management. The proposed protocol is designed to meet a required level of security (e.g., the fraction of valid, correct and uncompromised public keys, and information risk) as well as to meet performance requirements (e.g., service availability and communication overhead), without relying on trusted third parties such as CAs. The proposed protocol aims to achieve: (a) resiliency against misbehaving nodes in the network to maintain minimum security vulnerability; (b) availability in service provision in the presence of compromised nodes; and (c) efficiency in minimizing communication overhead incurred by trust and key management operations. CTPKM satisfies the requirements of self-organized and distributed key management for MANETs as discussed in [14]: (a) no single point of failure, i.e., no trusted third party is required; (b) resilience with low security vulnerability in the presence of hostile entities, i.e., little exposure of a compromised key; (c) high service availability, i.e., a sufficient number of valid, correct public keys are kept in each node; and (d) scalability, i.e., low communication overhead for obtaining a valid/correct public key whose corresponding private key is not compromised.

The contributions of our work are as follows:

1. Relative to existing non-trust-based distributed key management algorithms for MANETs without using a centralized trusted [15–19], our contribution is to develop a CTPKM that allows each node to make local peer-to-peer trust assessment for distributed decision making based on a composite trust metric. We consider multiple dimensions of trust (i.e., competence, integrity, and social contact) that are estimated based on evidence derived from the characteristics of communication, information, and social networking in a MANET. This allows fast and safe propagation of the keys to trustworthy nodes for preserving quality-of-service (QoS).
2. Relative to existing trust-based distributed key management algorithms for MANETs without using a centralized trusted CA [20–24], our contribution is to develop a threshold-based filtering mechanism that can effectively exploit the inherent tradeoff between trust and risk. The end result is that CTPKM is able to identify the optimal trust threshold to be applied at runtime for differentiating trustworthy vs. untrustworthy nodes to maximize key management service availability.
3. We conduct a comprehensive performance analysis comparing CTPKM with both non-trust-based and trust-based counterparts. We demonstrate that CTPKM outperforms a non-trust-based baseline model and two existing trust-based key management schemes [20,21], and can identify an optimal operational setting meeting dual conflicting goals of performance and security.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the system model including the attack model, trust model, protocol description, and performance metrics. Section 4 conducts a comparative performance analysis and reports numerical results. Section 5 concludes our paper and suggests future work.

2. Related work

In this section, we discuss existing work in certificate-based public key management for MANETs, and compare

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

and contrast them with our proposed CTPKM (Section 2.1). For completeness, we also survey identity-based (Section 2.3), threshold-based (Section 2.2), certificate-less cryptography (Section 2.4), and combined (Section 2.5) public key management, and provide reasons why these other approaches (Sections 2.2–2.5) are not suitable for public key management for MANETs.

2.1. Certificate-based public key management

Certificate-based public key management approaches require public keys to be distributed where the receiving party should be able to authenticate the received key based on the certificate of the public keys. Thus, a trusted CA is required to deal with key management operations including key generation, distribution, revocation, and update [25].

For MANETs without trusted CAs, certificate-based approaches should operate in a self-organized way. Capkun et al. [15] proposed a certificate-based self-organized public key management for MANETs by removing the need of a centralized trusted entity for key management. However, the assumption of being able to create certificate graphs is unrealistic as MANETs suffer from unreliable wireless transmission, high security vulnerability, and dynamically changing topologies. The criteria being used by a user to issue a public-key certificate of another user are not provided, even though the existence of a public key is used as evidence to trust another node. Further, the claimed benefit of low communication cost in the presence continuous updates of certificate repositories of users in dynamic, hostile MANETs is questionable. The authors [20,24] proposed a two-step secure authentication protocol for multicast MANETs. In order to deal with key management, they used the highest trustworthy node as a CA and the second highest trustworthy node as a backup CA. However, the measurement of trust values is not clearly described. Chauhan and Tapaswe [16] proposed a key management approach for MANETs with no trusted third entity. This work employs a group leader as a CA to manage key generation and distribution. However, group leader selection is done randomly, without considering its trustworthiness and specific attack behaviors. Dahshan and Irvin [21] proposed a certificate-based public key management scheme for MANETs where trust is used as authentication metric to represent the assurance of obtaining a valid public key. However, this scheme generates high communication overhead and delay for a source to obtain a valid public key of a destination.

Huang and Wu [17] proposed a certificate path discovery algorithm for MANETs based on the hierarchical PKI structure using multiple CAs with no specific trust framework. Huang and Nicol [18] proved that the shortest certificate chain does not guarantee the most trustworthy path to obtain the public key of a target node due to different trustworthiness observed in each intermediate node on the certificate chain. In order for public keys and their certificates to be managed by trustworthy CAs in the hierarchical public key management structure, Xu et al. [24] and PushpaLakshimi et al. [22] used cluster heads as trustworthy CAs based on their trust in the system. However, these works [22,24] still reveal a single point of failure and did not show specific trust models and attacks considered. Wu et al. [19] proposed a key management protocol for MAENTs for efficiency in updating certificates and security in key revocation. They used a special group named "server group" consisting of servers of multicast groups. However, the selection algorithm of the servers in the server group is not discussed. Vinh et al. [23] employed a group head for public key management in a group communication system where the group head is selected based on trust. However, this work does not detail the used trust mechanism.

Many studies used certificate-based public key management. However, they have brought out practical limitations including high communication overhead or delay [15,21], and using static trust [20,22–24] to select CAs. In addition, hierarchy-based selection of CAs (e.g., group leaders or cluster heads) [16–19] also reveals high security vulnerabilities when the selected CAs are compromised. Our work differs from the above works in that we devise a trust metric considering multiple dimensions of a node's trust and leverage it for decision making in the process of key management and group communication in order to achieve system goals including high service availability, low communication cost, and low risk.

Existing certificate-based public key management schemes cited above expose practical limitations, including needing a centralized trusted CA [25], high communication overhead or delay [15,21], and using static trust [20,22–24] to select CAs. In addition, hierarchy-based selection of CAs (e.g., group leaders or cluster heads) [16–19] can lead to high security vulnerability when the selected CAs are attacked and compromised.

Unlike [25] our work does not use a centralized trusted CA. Unlike [16–19] cited above, our work uses a composite trust specifically designed for public key management. Peer-to-peer trust evaluation is dynamically performed over time upon interactions between entities. This novel design feature contributes to (1) detecting malicious entities that have been compromised over time; and (2) issuing/distributing a public/private key pair to only nodes that maintain a certain level of trust. This feature also mitigates the security vulnerability issue suffered by hierarchy-based CA selection schemes [16–19] . Unlike[15,21] which incur high communication overhead or delay, we develop a threshold-based filtering mechanism that can effectively exploit the inherent tradeoff between trust and risk in order to achieve system goals including high service availability, low communication cost, and low risk.

2.2. Threshold public key cryptography

Shamir [26] proposed *threshold cryptography* based on sharing of secrets to generate a private key. In *threshold cryptography*, the private CA key is distributed over a set of server nodes through a (k, n) secret sharing scheme. The private CA key is shared between n nodes in such a way that at least k nodes must cooperate in order to sign the certificates. However, a central trusted CA exists to select servers as the coordinators for key management, resulting in a single point of failure. In addition, the inherent weakness of the secret sharing scheme is the substantial

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

delay when the set of trustworthy server nodes cannot be found to generate the private CA key [27,28]. Besides, when the CA is compromised, the whole system is compromised [29].

2.3. ID-based public key cryptography

Shamir [30] also proposed the concept of *ID-based public key cryptography* (ID-PKC) which generates a public key based on the ID of the node (e.g., IP or email address) and its corresponding private key generated by a trusted CA. The weakness of the ID-based scheme is well-known as a *key escrow problem* which exposes high security vulnerability when the trusted CA is compromised. To remove the key escrow problem, several solutions have been proposed including ID-based authentication schemes [31,32], secure private key generation using simple blinding technique in pairing-based cryptography [33]. This approach is popularly applied in resource-restricted network environments [34] due to low communication overhead by reducing the size of secret information (i.e., ID) to generate a public key. However, these works assumed the existence of a trusted entity (or entities) to issue or coordinate public/private key pairs. This reveals high security vulnerabilities when the trusted coordinators are compromised. In particular, no trust evaluation is considered to reflect dynamic status of trust in entities where a node can be compromised over time.

2.4. Certificateless public key cryptography

To cope with the communication overhead incurred in exchanging certificates, the concept of certificateless public key cryptography (CL-PKC) is introduced [35]. CL-PKC is a variant of ID-PKC devised to prevent the key escrow problem in ID-PKC. CL-PKC uses a trusted third party (TTP) which generates a partial private key to an entity based on a master key and the entity's ID. The entity then generates its actual private key based on the partial private key provided by the TTP and its secret information. By this way the TTP cannot access the private key of an entity. Compared to traditional public key cryptographic systems, CL-PKC does not require the use of certificates to ensure the authenticity of public keys, leading to less communication cost generated. Due to this lightweight feature, CL-PKC has been used for securing MANETs [36–38]. However, this cryptography reveals security vulnerability in that an attacker can fake a user's public key because the part of the user's public key is from the user's random secret value.

2.5. Hybrid public key management

Some researchers proposed hybrid public key management mechanisms that combine the features of multiple schemes to meet the requirements. Sun et al. [39] combined ID-based key management with threshold cryptography without using a centralized third party to deal with key management. Xu et al. [24] combined certificateless public key cryptography which eliminates the key escrow problem with threshold cryptography which does not require a centralized third party. Zhang et al. [36] proposed an ID-based key management scheme that combines ID-based cryptography with threshold cryptography to enhance security and reduce communication cost for key management. Li and Liu [40] also proposed a hybrid key management scheme combining ID-based key management and threshold cryptography.

All the hybrid schemes cited above [24,36,39,40] could not completely remove the need of a centralized trusted authority because ID-PKC has an inherent escrow problem and the threshold cryptography requires a trusted third authority to select trustworthy multiple key servers that generate the secret shares of a private CA key. If the trusted entity is compromised, the entire system will be vulnerable. In addition, the use of threshold cryptography can cause a high delay when generating a key because a sufficient number of multiple trustworthy servers may not be available in dynamic MANETs.

3. System model

We consider a MANET with no centralized trusted CA that deals with public key management. Nodes are devices carried by a human entity (e.g., a soldier carrying mobile devices), modeled with heterogeneous characteristics with different monitoring capability, which affects detection errors, group join/leave rates, and different trust levels. To reflect real human mobility patterns in a MANET, we used CRAWDAD human mobility trace data collected by KAIST, Daegeon, Korea [41]. In the mobility data set, the locations of 92 human nodes over the university campus of KAIST were traced by GPS readings per 30 s for a day.

In this work, a public key may have the following status:: (1) *valid/invalid*: a key that is expired or not yet expired; (2) *correct/incorrect*: a key that is genuine or fake; and/or (3) *uncompromised/compromised*: a key whose corresponding private key is compromised or not. The status of the public key can belong to more than one category among three while each category gives a binary status.

3.1. Attack model

We assume if a node is compromised, the node can perform random attacks [6] with an attack intensity probability (P_a) to evade detection. Attack intensity is modeled by a probability parameter, P_a, specifying the frequency of triggering attacks by an attacker. We consider the following attacks in MANETs:

- *Packet dropping*: A node may drop a packet received due to the nature of selfishness (e.g., to save energy) or maliciousness (e.g., to interrupt service availability). This is detected by overhearing to see if a packet sent to a neighbor for forwarding is actually being forwarded. It is not possible to tell if packet dropping is a problem of competence or integrity. Given that there are many attack behaviors that can be detected by our protocol design to attribute to integrity, to avoid double-count we simply attribute packet dropping to competence. If a node drops packets and the behavior is observed by a neighbor, this neighbor will decrease the misbehaving node's direct competence trust. Furthermore, this neighbor when acting as a recommender

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

will propagate a negative recommendation to other nodes as indirect evidence against the misbehaving node. See Section 3.2.3 for the detail of peer-to-peer trust estimation based on the aggregation of both direct and indirect evidence.

- *Private key compromise*: A node's private key compromise can occur in the two ways: (1) when the node itself is compromised and passes its private key to other compromised nodes or outside attackers to leak confidential information out; and (2) when a public key certifier (called a neighborhood trustworthy certifier, denoted as NTC, to be defined in Section 3.3.1) is compromised and leak out a private/public key pair of the victim node to outside attackers. The outside attackers can impersonate the victim and obtain access to confidential information that should be shared only by group members. This attack can be detected based on majority voting in our protocol design (see Section 3.3.3). When a private key compromise attack is detected, the public/private key pairs of the victim or compromised node will be denounced. If an attacker continues to use the denounced public/private key pairs, it will be detected and the detection will attribute to lowering the attacker's trust in integrity. Using the trust threshold, a node will decide whether to believe the received public key is valid, correct, and/or compromised based on the perceived trust of the source sending the compromised public key.
- *Message modification/forgery*: A node may modify/forge a message received, hindering effective communication, and/or accurate trust assessment. This attack occurs when the attacker possesses the private key of the receiver due to private key compromise. However, when the corresponding private key compromise attack is detected (explained above), the public/private key pairs of the sender will be denounced. If an attacker continues to use the private key to do message modification/forgery attack, it will be detected and the detection will attribute to lowering trust in integrity.
- *Fake identity/impersonation*: A node may use a fake identity or multiple identities (i.e., Sybil attack) to break information confidentiality in communications between two entities. In particular, a node can impersonate as a victim node whose private key is compromised by distributing the public key and the certificate of the victim node to its neighbors in order to attract the victim node's packets to it. However, when the corresponding private key compromise attack is detected (explained above), the public/private key pairs of the victim node will be denounced. If an attacker continues to use the private/public key pairs to do fake identity attack, it will be detected and the detection will attribute to lowering trust in integrity.
- *Fake recommendation dissemination*: A node may give a bad recommendation for a good node while giving a good recommendation for a bad node in order to deter accurate trust evaluation/decision making. In our trust metric, recommendations are used as indirect evidence which may be delivered through multiple hops in MANETs. The correctness of the recommendations is ensured by unanimous agreement of the intermediate nodes based on their opinions towards the previous forwarding node (i.e., whether the node is lying or not) in the route in which the opinions are tagged in the main message delivered. The receiver can detect fake information dissemination attack by checking if there is a negative opinion for an intermediate node on the path. If yes, this detection will attribute to low trust in integrity (see Section 3.2.3 for more details).
- *Denial-of-service (DoS)*: A malicious node can generate unnecessary traffic to interrupt service provision in the system. We considered the DoS attack within the key management framework. Specifically, a malicious node can keep requesting public keys of other nodes even if it already has their valid public keys. Since only trustworthy nodes based on the trust threshold criterion are able to issue, distribute, and obtain key pairs, this DoS attack can consume network resources to increase delay of system operations, and reduce service availability. This attack is countermeasured by using a trust threshold for intermediate nodes to ignore public key requests generated from a node whose trust level is below the threshold, thus effectively throttling DoS attacks.
- *Whitewashing*: A malicious node may leave a network and come back later with a new reputation. In our trust management protocol, all new nodes joining a network will start with a trust value of ignorance (i.e., 0.5), when other nodes assess their trust upon join. For a new node joining the network, it is allowed to interact with other nodes to accrue reputation from other nodes with a given warming-up period. If after this period, the new node does not reach the trust threshold, it will be isolated from group activities and cannot obtain a valid key pair. Once a new node accrues its reputation over time through interactions with other nodes or observations by direct neighbors, the increased trust enables the new node to participate in key management operations.

We summarize how each attack is detected and countermeasured by the design features of CTPKM in Table 1. Table 2 summarizes the attack behaviors during the operation of our key management protocol.

3.2. Trust model

3.2.1. Dimensions of trust

We consider three trust components to capture the unique aspects of trust in a MANET with communication, information and social networking:

- **Competence (C)** refers to an entity's capability to serve requests in terms of a node's cooperativeness and availability. Availability may be affected by network conditions such as link failure, energy depletion, and voluntary or involuntary disconnection (i.e., leaving the network). This is measured by the ratio of the number of positive experiences to the total experiences in packet forwarding.
- **Integrity (I)** is the honesty of an entity in terms of attack behaviors discussed in Section 3.1 except packet dropping behavior. This is measured by the number of

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

Table 1
Attack behavior, detection, and countermeasures.

Attack behavior	Detection	Countermeasure
Packet dropping	Overhearing by a monitoring mechanism pre-installed in each node	Lowering direct trust in competence by neighboring nodes (direct evidence); propagation of low competence to other nodes via recommendations (indirect evidence)
Private key compromise	Majority voting by neighboring nodes who detect the private key compromise based on the attacker's integrity	Lowering the attacker's integrity by neighboring nodes
Message modification/forgery	Majority voting by neighboring nodes who detect the private key compromise based on the attacker's integrity	Lowering the attacker's integrity; propagation of the low integrity via recommendations
Fake identity/Impersonation	Checking the integrity of both the owner and issuer (i.e., NTC) of the private/public keys based on trust threshold, T_{th}	Lowering the culprit's integrity; propagation of the low integrity via recommendations
Fake recommendation dissemination	A recommendation packet delivered that does not have unanimous agreement of positive opinions by all intermediate nodes on the path	The recommendation is discarded; lowering the attacker's integrity; propagation of the low integrity via recommendations
Denial-of-service	Receiving a large amount of the same requests from a node whose integrity trust is below T_{th}	Lowering the attacker's integrity by neighboring nodes; propagation of the low integrity via recommendations
Whitewashing	After a warming-up period allowing a new joining node started with ignorance trust (0.5) to interact with other nodes, the node's trust does not reach T_{th}	The new joining attacker with trust less than T_{th} cannot have a valid pair of its own private/public key

Table 2
Attack behavior for operations.

Operation	Attack behavior
Trust assessment	Fake information dissemination, message modification, packet dropping
Key issuance by a malicious key generator	Private key compromise
Public key distribution	Compromised public key distribution, fake identity
Public key request delegation	Packet dropping, message modification, identity impersonation
Forwarding a requested public key	Message modification/forgery by forwarding a fake public key
Network join	Whitewashing

positive experiences over the total experiences related to protocol compliance.

- **Social contact (SC)** is defined based on a node's inherent sociability derived from the trust profile available a priori as well as dynamic social behavior measured by the number of nodes that a node encounters during a trust update interval T_u over the total number of nodes in the network. If an entity has high SC, it is more likely to disseminate information quickly to the network, compared to the ones with low SC. An entity's mobility pattern will affect this trust component.

In this work, trust is used for making decisions, including obtaining a certificate of a public key, distributing a public key, requesting a public key of a target node, and providing a public key requested. The reasons we pick the above three trust components are (1) with *competence trust*, we assure fast propagation of public keys; (2) with *integrity trust*, we increase the probability that public keys propagated are valid/correct with the corresponding private key uncompromised; and (3) with *social contact trust*, we increase the probability of finding a valid public key from nodes having good social networking.

Henceforth, we denote the trust values of node i towards node j in trust component X's (= competence, integrity, and social contact) at time t by the notations of $T^{C}_{i,j}(t)$, $T^{I}_{i,j}(t)$, and $T^{SC}_{i,j}(t)$. We follow the trust computation model in our prior work [10] to assess $T^{X}_{ij}(t)$ at time t.

3.2.2. Objective trust

We assume that a node's trust profile is available, describing its inherent behavior patterns that can be scaled in [0, 1]. In this work, we generate a node's initial average trust value, called its trust seed, from $U(GB, 1)$, where U is a random real number generator function based on uniform distribution with the lower and upper bounds as input and GB is the lower bound for good behavior. There is a separate trust seed for each trust component X, where $X = C,\ I$ or SC for competence, integrity, or social contact respectively. Let S^{X}_{i} denote the trust seed drawn from $U(GB, 1)$ for trust component X of node i. Let $P^{X}_{i}(t)$ be the actual trust seed drawn from $U(S^{X}_{i} - P_d, S^{X}_{i} + P_d)$ at time t with S^{X}_{i} being the mean and P_d being the standard deviation of a node's average behavior from its actual behavior to account for behavior variation as a function of time. Then,

$$P^{X}_{i}(t) = \min[U(S^{X}_{i} - P_d, S^{X}_{i} + P_d), 1] \quad (1)$$

$$S^{X}_{i} = U(GB, 1) \text{ for } X = C, I \text{ or } SC \quad (2)$$

Let $T^{I}_{i}(t)$ denote the "ground truth" objective trust of node i at time t. For $X = C$ (for competence), we have to account for node availability. Hence, $T^{C}_{i}(t)$ is calculated by the product of $P^{C}_{i}(t)$ with P_r, where P_r is the link reliability considered for competence trust at time t, as follows:

$$T^{C}_{i}(t) = P^{C}_{i}(t)P_r \quad (3)$$

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

For $X = I$ (for integrity), $T_i^I(t) = P_i^I(t)$ if node i is not compromised at time t. If node i is compromised at time t, $T_i^I(t)$ depends on how often node i preforms attacks. We assume that a compromised node will perform random attacks (on-off attacks) with probability P_a (attack intensity) to evade detection. If the compromised node does not perform attack at time t, its integrity trust $T_i^I(t)$ is equal to $P_i^I(t)$ because it is not detectable. If the compromised node performs attack at time t, its integrity trust $T_i^I(t)$ is decremented by P_a from the past trust value $T_i^I(t - \Delta t)$ with a lower bound of zero. Note that in CTPKM, we notate Δt with a trust update interval, T_u. Summarizing above, for a compromised node, $T_i^I(t)$ is given by:

$$T_i^I(t) = \begin{cases} P_i^I(t) \text{ if node } i \notin C \\ \max[T_i^I(t - \Delta t) - P_a, 0] \text{ if node } i \in C \end{cases} \quad (4)$$

Here C is the set of malicious nodes performing attack at time t and Δt is the periodic trust update interval T_u. In our experiment, we set the percentage of nodes to be compromised at $t = 0$ based on P_c to test the resiliency of our protocol against increasing malicious node population.

Lastly when $X = SC$ (for social contact), we have to account for node dynamic social behavior, so $T_i^{SC}(t)$ is given by the product of $P_i^{SC}(t)$ with $\rho N_i^e(t)$, where $N_i^e(t)$ is the number of encounters to node i as 1-hop neighbors during the previous T_u, and ρ is , as follows:

$$T_i^{SC}(t) = \rho P_i^{SC}(t) N_i^e(t). \quad (5)$$

3.2.3. Subjective trust

Node j's trust values for component X's evaluated by node i is computed based on the aggregation of both direct and indirect evidences. Trust of node j (trustee: trusted party) evaluated by node i (trustor: trusting party) in trust component X is:

$$\text{if } (|R_j| > 0) \wedge (HD(i,k) \geq TC) \quad (6)$$

$$T_{i,j}^X(t) = \alpha T_{i,j}^{D-X}(t) + (1-\alpha) T_{i,j}^{ID-X}(t);$$

$$\text{else } T_{i,j}^X(t) = \gamma T_{i,j}^X(t - \Delta t);$$

$T_{i,j}^{D-X}(t)$ is direct trust based on direct observations and $T_{i,j}^{ID-X}(t)$ is indirect trust based on recommendations from 1-hop neighbors of node j. R_j is a set of recommendations correctly received from node j's 1-hop neighbors. The availability of recommendations ($|R_j| > 0$) is affected by the receipt of the correct recommendations that are affected by packet dropping behaviors and integrity (i.e., attack behaviors) of a node. $HD(i, k)$ is the number of hop distances between nodes i and k where node i is a requestor (the trustor) and node k is a 1-hop neighbor and recommender of node j (the trustee).

In order for the new indirect evidence to be used for trust update, recommender node k for node j should exist within TC hops from node i and the correct recommendation should arrive safely at node i where TC is the maximum number of hop distances, called a trust chain, allowed to collect recommendations from 1-hop neighbors of node j. We will use the optimal TC identified in this work, but do not investigate this in detail because this is already examined in [10]. The correctness of the recommendations is ensured by referring to direct opinions of referral recommenders (forwarding the original recommendation) attached to the original message with any detection error of the intermediate nodes forwarding the recommendation [10].

In (6), α and $1 - \alpha$ are the weights for direct and indirect evidence respectively where α is set between 0 and 1. We observe that when the weight (α) for direct evidence is low, high trust accuracy is observed, or vice-versa. This is because only correct recommendations based on unanimous agreement by all intermediate nodes passing the recommendation are used as indirect evidence while new direct evidence cannot be collected easily due to node mobility. When no correct recommendations are received from recommenders k's located within TC hops from node i and this is detected by node i based on the direct opinions of intermediate nodes attached to the recommendation, trust decays with a decay factor γ over Δt, the periodic trust update interval T_u. However, due to the collusion of compromised nodes, detection errors can be introduced and false recommendations may be considered. In the trust metric used in this work, the false detection can be minimized by requiring the unanimous agreement of all intermediate nodes about the correctness of the recommendation delivered.

The direct trust (based on direct observations) of node j evaluated by node i on trust component X at time t, $T_{i,j}^{D-X}(t)$, is computed as:

$$T_{i,j}^{D-X}(t) = \begin{cases} P_{i,j}^{D-X}(t) \text{ if } HD(i,j) == 1 \\ \gamma T_{i,j}^X(t - \Delta t) \text{ otherwise} \end{cases} \quad (7)$$

When nodes i and j encounter as 1-hop neighbors (i.e., $HD(i, j) == 1$) during the time period $(t - \Delta t)$, node i can collect direct evidence based on its own observations or experiences $P_{i,j}^{D-X}(t)$. When nodes i and j are distant with more than 1 hop distances, node i relies on its past experience to assess the direct trust of node j. $P_{i,j}^{D-X}(t)$ for $X = C$ or I is computed based on the positive experience over the negative experience associated with X as:

$$P_{i,j}^{D-X}(t) = \begin{cases} \frac{r}{r+s} \text{ if } r + s > 0 \\ 0 \text{ otherwise} \end{cases} \quad (8)$$

where r is the number of positive experiences and s is the number of negative experiences.

$$P_{i,j}^{D-SC}(t) = \begin{cases} \rho P_j^{SC}(t) N_j^e \\ 0 \text{ otherwise} \end{cases} \quad (9)$$

$P_j^{SC}(t)$ is given from the available trust profile and N_j^e and ρ are explained in (5); N_j^e can be directly observed by 1-hop neighbors of node j. Note that 1-hop neighbors of node j are to forward recommendations to node i.

The indirect trust of node j evaluated by node i on trust component X at time t, $T_{i,j}^{ID-X}(t)$, is obtained by:

$$T_{i,j}^{ID-X}(t) = \begin{cases} \frac{\sum_{k \in R_j} T_{k,j}^X(t)}{|R_j|} \text{if } |R_j| > 0 \\ \gamma T_{i,j}^X(t - \Delta t) \text{ otherwise} \end{cases} \quad (10)$$

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

When node i receives correct recommendations with $|R_j| > 0$, node i uses the average of the recommendations to derive the overall indirect trust. If R_i is an empty set, node i will use its past experience $T^X_{i,j}(t - \Delta t)$, with decay weighted by γ, due to no correct recommendations received.

In this work, we use a trust threshold, T_{th}, for a node to make a routing decision with the goal of safe delivery of a message without being modified by untrustworthy nodes on a path the message travels [8]. This mechanism is applied when a recommendation is delivered from recommenders for a trustee to a trustor.

3.3. Composite trust-based public key management

In this section, we discuss the core operations of CTPKM as illustrated by Fig. 1. Each mobile entity is able to communicate with other entities using public/private key pairs obtained through CTPKM. Given a trust threshold T_{th}, a node will assume a certain amount of risk to communicate with another node whose trust level is no less than T_{th}.

3.3.1. Key generation

In CTPKM, each node generates its own public/private key pairs periodically but the key pair should be certified by a trusted third party which generates the certificate of the public key. Since CTPKM does not assume the existence of a trusted third party, each node needs to find the most trustworthy third party node among its 1-hop neighbors, called *neighborhood trustworthy certifier* (NTC) which can certify the self-issued private/public keys. The reason a node chooses NTC among its 1-hop neighbors is due to resource constraint and security vulnerability in MANETs which do not allow trusted third parties. This certification process mitigates a compromised node to obtain its public/private key because NTC issues the certificate only when the requesting node is evaluated as trustworthy based on whether NTC's trust in the requesting node is no less than a given trust threshold. Now we discuss how NTC issues a certificate to a node requesting the certificate of its public/private key pair.

3.3.2. Public key certificate issuance

Each node i asks NTC m, a node having the highest trust value among i's 1-hop neighbors, to certify the public key it generates. The minimum condition to be an NTC is that the NTC must have at least a trust value no less than the given trust threshold (T_{th}) for integrity trust (i.e., $T^I_{i,m}(t) \geq T_{th}$). Thus, if i cannot find any 1-hop neighbors who have a trust value no less than T_{th}, it cannot obtain a valid key pair.

After an NTC, m, receives i's request for the certification of i's key pair, it decides whether to issue the public key certificate based on i's trustworthiness, in integrity trust using T_{th} (i.e., $T^I_{m,i}(t) \geq T_{th}$). That is, there should be a mutual trust relationship between a certificate requestor and an issuer in integrity trust. The requesting node i will not be able to obtain the certificate of its public key if its integrity trust level is below T_{th}. Recall that trust values are dynamically changing over time. The trust threshold T_{th} affects the protocol performance as follows. If a low T_{th} is used, even a relatively untrustworthy node can issue and certify the public key to others. As a result, a malicious NTC can disseminate many fake public keys so as to generate unnecessary communication, resulting in a waste of network resources. Besides, a malicious NTC who generates the private key can perform private key compromise attacks and intercept information which is sent to the originally intended recipient. If either the NTC or the victim node of the public key is compromised, then the compromised NTC or the victim node can leak the private key to other attackers as well. Hence, using an optimal trust threshold for all decisions associated with key management is critical to mitigating the security vulnerability.

NTC also issues an expiration time of the new key pair where the expiration time-stamp is part of the certificate. A node updates its public/private key pair when the expiration time is reached or when its private key is detected as compromised. Intuitively, the longer the expiration time the lower the security, revealing high security vulnerability. However, the longer expiration time allows lower communication cost. We assume a fixed expiration time for all nodes' certified key pairs in this work.

3.3.3. Public key distribution

After a node obtains its public key certificate, the node disseminates the public key along with the certificate to a subset of its 1-hop neighbors whose trust values are no less than T_{th} for all three trust components. That is, node i will disseminate its public key packet to a neighbor m which should meet the following conditions:

$$T^X_{i,m}(t) \geq T_{th} \tag{11}$$

where $T^X_{i,m}(t)$, with $X = C, I$ or SC, is a measured trust of node m evaluated by node i for competence, integrity, or social contact trust, respectively. Node i also periodically disseminates its public key to its current 1-hop neighbors who satisfy the above conditions.

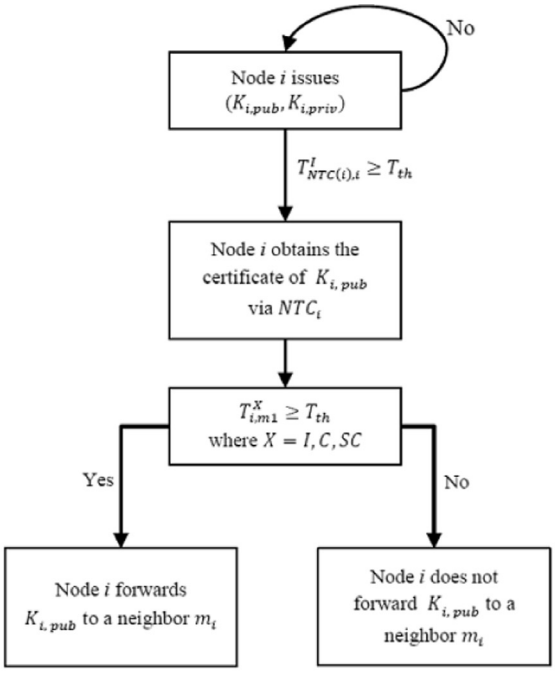
Since nodes are mobile, if a node has a high mobility rate, it may have more chances to obtain public keys of other nodes (being affected by the degree of social contact trust), and vice-versa. Selecting the right set of neighboring nodes is critical to revealing less security vulnerability while obtaining uncompromised public keys. When a public key is distributed to a compromised node, the compromised node may perform a data forgery or modification attack by forwarding a fake pubic key.

When node i disseminates its public key and the certificate to a subset of 1-hop neighbors based on (11) (called "trustworthy 1-hop neighbors" hereafter), the packet consists of the following items:

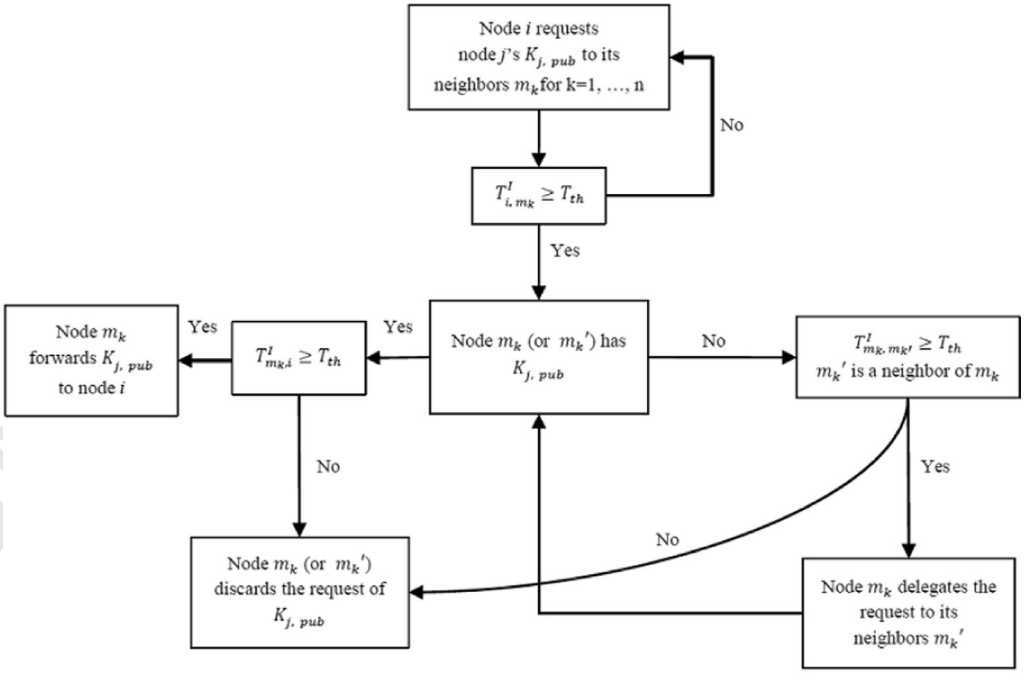
$$[C^{NTC_i}_{K_{i,pub}}, K_{i,pub}] \tag{12}$$

$C^{NTC_i}_{K_{i,pub}}$ is the certificate of node i's public key signed by the NTC's digital signature including the information on the node ID (node i), the NTC's ID, and expiration time for the valid period of the public key. Note that the notation NTC_i is the NTC who issues the certificate of node i's public key and $K_{i,pub}$ is node i's public key. The receiving node will check the trustworthiness of the NTC in integrity trust

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014



(a) *Key issuance and distribution*



(b) *Key request*

Fig. 1. Distributed key management process in CTPKM.

with T_{th} to ensure the authenticity of $K_{i,pub}$. If the NTC is compromised, it can perform a private key compromise attack by leaking the private key generated to other malicious nodes. Integrity trust check of the NTC minimizes the chance of this attack. Node i distributes this information to its selected 1-hop neighbors in the clear, in as specified in (12), because node i may not know the public keys of its neighbors upon entry. This may reveal the vulnerability to

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

message forgery attacks by which a malicious node may intercept the information and send a fake key and certificate to the intended receivers. To minimize this vulnerability, our protocol design ensures the authenticity of the disseminated public key and certificate via the agreement of the receivers towards the opinions of the authenticity of the key pair based on a majority voting mechanism (i.e., if the majority of voters agree, then the public key and certificate are considered authentic). Unless more than a majority of the receivers are compromised, this ensures the authenticity of the received key and certificate. We consider the extra communication overhead caused by this authenticity check in our performance metric.

Some nodes may not have the public key of a particular node it wants to communicate with because it has not encountered the node as a 1-hop neighbor. In this case, a node can request the target node's public key from its trustworthy 1-hop neighbors based on (11). If any of the trustworthy 1-hop neighbors (m) has the public key of the target node (TN), then it will provide the public key to node i. Whether or not to provide the public key of the TN depends on m's assessment toward node i (requestor) in integrity trust (i.e., $T^I_{m,i}(t) \geq T_{th}$). If node m decides to provide the public key of TN, the returning message includes the following items:

$$[C^{NTC_{TN}}_{K_{TN,pub}}, K_{TN,pub}, ID_m]_{K_{i,pub}} \quad (13)$$

Node m returns the public key of the TN, $K_{TN,pub}$, the TN's public key certificate $C^{NTC_{TN}}_{K_{TN,pub}}$, and its ID ($ID_m$). The message is encrypted by $K_{i,pub}$, the public key of node i. When the requestor receives this message, it will save the public key of TN if m satisfies (11).

If m does not have the public key of the TN, it will forward the request message to its trustworthy 1-hop neighbors (m''s) that meet (11). The delegated request message has the following format:

$$[C^{NTC_i}_{K_{i,pub}}, K_{i,pub}, ID_{TN}]_{K_{m',pub}} \quad (14)$$

$C^{NTC_i}_{K_{i,pub}}$ is the public key certificate of node i, certified by the NTC of node i, NTC_i. ID_{TN} is the ID of the TN, and $K_{m',pub}$ is the public key of node m' who is a trustworthy 1-hop neighbor of m. Node m' receiving the delegated request message from m decrypts the message with its private key, checks if it has the public key of the TN, and checks if the requestor, node i, passes the integrity test (i.e., $T^I_{m',i}(t) \geq T_{th}$). If yes, then it sends the public key of TN to the original requestor (node i) by a returning message following the format specified in (13).

If an intermediate node forwarding the request message is uncooperative, the request message can be dropped; therefore, many nodes may not have valid public keys. A malicious 1-hop neighbor may even provide incorrect public keys. Therefore, the trust threshold (T_{th}) affects how many valid, correct and uncompromised public keys a node can use. In our proposed CTPKM, we show there exists an optimal T_{th} under which a sufficient number of valid, correct and uncompromised public keys are generated while reducing communication overhead (not forwarding the public key requests to untrustworthy nodes) and mitigating security vulnerability (reducing the use of a compromised public key). In CTPKM, when a trustworthy intermediate node that meets (11) has a valid public key of the TN, it can provide the public key of the TN to the requestor. This reduces communication overhead significantly in key distribution.

Key revocation and update: The private/public keys of a node will be revoked after the valid period expires. Since the certificate includes the information on expiration time, key revocation due to time expiration will be implicitly known to other nodes in the network. Before the valid period is past, a node's 1-hop neighbors can serve as verifiers and apply majority voting to detect if the node's private key is compromised. If a verifier had once interacted with another node claiming it to be the target node, then the verifier suspects the target node's private key has been compromised, and will vote against it. If the private key is deemed compromised (when the majority of the neighbors vote against it), the node, being as the owner of the private key, must notify the key compromise event to all nodes in the network. We consider the extra communication overhead caused by this key revocation procedure in our performance metric. If the owner of the key itself is compromised and does not disseminate the key compromise message, its neighbors will decrease the trust value of the node to hinder the node from reissuing a new public/private key pair. In CTPKM, if a node does not maintain a certain level of trust, it cannot obtain the certificate of its public key. Therefore, there is a very low chance for an untrustworthy node to issue its public key with a valid certificate.

3.4. Metrics

To examine the impact of attack intensity and ratio of attackers on trust accuracy, we define the following metric:

- **Trust bias** ($\mathcal{B}^X$) refers to the difference (absolute value) between ground truth trust values and estimated trust values in trust property X. In this paper, we apply the optimal trust chain length (TC) protocol design [10] explained in Section 3.2 to minimize the trust bias. Given the optimal TC, the trust bias of node j evaluated by node i is measured by:

$$\mathcal{B}^X = |T^X_i(t) - T^X_{i,j}(t)| \quad (15)$$

$T^X_i(t)$ indicates the ground truth trust value of node i in trust component X, as explained in Section 3.2.2. The measured trust of node i evaluated by node j, $T^X_{i,j}(t)$, is computed in Equations (6)–(10).

CTPKM is built on top of the optimal TC protocol design and is measured by the following performance metrics:

- **Fraction of correct public keys** ($\mathcal{F}$) refers to the average number of valid, correct and uncompromised public keys kept in each node over the total number of member nodes in the network, computed by:

$$\mathcal{F} = \frac{\sum_{t=0}^{LT} \sum_{i \in M} \sum_{j \in M, j \neq i} K_{i,j}(t)}{|M||M-1|LT} \quad (16)$$

where $K_{i,j}(t) = 1$ if node i has the valid, correct and uncompromised public key of node j; 0 otherwise. M is a

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

set of legitimate members in the network. The entire mission time is denoted as LT (lifetime).

- **Service availability** ($\mathcal{A}$) refers to the ratio of the average time period that a node's valid, correct and uncompromised public key is kept by other nodes over the entire session time, calculated by:

$$\mathcal{A} = \frac{\sum_{i\in M}\sum_{j\in M, j\neq i} A_{i,j}}{|M||M-1|LT} \tag{17}$$

where $A_{i,j}$ is the time duration node i has the valid, correct and uncompromised public key of node j. This metric implies how much time each node keeps another node's valid, correct and uncompromised public key during the entire session time.

- **Information risk** ($\mathcal{R}$) indicates the average number of packet transmissions using a compromised public key during the entire session, LT . Consider that each node sends a message to another node for which it keeps the public key in every group communication interval (T_g). The information risk exposed by sending messages using a compromised public key is computed by:

$$\mathcal{R} = \frac{T_g}{LT}\sum_{t=0}^{LT}\sum_{i\in M}\sum_{j\in C, j\neq i} R_{i,j}(t) \tag{18}$$

where $R_{i,j}(t) = 1$ if node i keeps a compromised public key of node j; 0 otherwise. C is the set of legitimate members whose private keys are compromised. If a node's private key is detected as compromised, the node is prohibited from group communication until its key is re-issued.

- **Communication cost** ($\mathcal{C}$) counts the number of hop messages per time unit (i.e., *second*) caused by CTPKM, computed by:

$$\mathcal{C} = \frac{\sum_{t=0}^{LT} C_{total}(t)}{LT} \tag{19}$$

with

$$C_{total}(t) = C_{te}(t) + C_{km}(t) + C_{gc}(t)$$

$$C_{km}(t) = C_{ki}(t) + C_{kd}(t) + C_{kr}(t)$$

$C_{te}(t)$ is the number of hop messages caused by trust evaluation accounting for the cost for each node to periodically (in every T_u) disseminate the trust values of its 1-hop neighbors to nodes located within the trust chain length (TC) [10]. $C_{km}(t)$ is the number of hop messages caused by key management. $C_{gc}(t)$ is the cost for group communication by all member nodes. $C_{km}(t)$ consists of three cost components: key issuance ($C_{ki}(t)$), key distribution ($C_{kd}(t)$), and key revocation ($C_{kr}(t)$). $C_{kd}(t)$ includes the cost for a public key to be distributed to trustworthy 1-hop neighbors and requesting nodes, and the cost for authenticating the public key by 1-hop neighbors.

4. Simulation results

This section shows numerical results obtained from simulation. We first explain the experimental setup and schemes to be compared against the proposed CTPKM. Then we conduct a comparative performance experiments and demonstrate numerical results with analysis.

4.1. Experimental setup

Our simulation is conducted using an event driven simulator SMPL [42]. Table 3 gives the set of parameters and their default values for defining the simulation environment. We use the optimal trust chain length (i.e., $TC = 4$) and $(\alpha, \gamma) = (0.1, 0.95)$ where α is a weight for direct evidence, $1-\alpha$ is a weight for indirect evidence, and γ is a decay factor. This setup environment is used for maximizing trust accuracy (or minimizing trust bias) while maintaining acceptable communication overhead due to trust assessment. The optimal TC and (α, γ) parameter settings are determined following our prior work [10,43] and the detail is not repeated here.

To model the mobility patterns of nodes in a MANET, we use CRAWDAD human mobility traces collected by KAIST [41] with $N = 92$ nodes. To ensure the availability of 92 nodes' mobility data, we take the initial 4 h of mobility data in order to trace available locations of all 92 nodes. A node may leave or join the network with the interval of $1/\mu = 4$ h and $1/\lambda = 1$ h, respectively. Due to sparse network connectivity, we scale down the operational area in order for nodes to have a sufficient level of interactions based on the radio range given ($R = 250$ m).

Initial values for each trust component are seeded with a random variable selected from the range in [GB, 1] based on uniform distribution where GB is the lower bound. The trust values are also affected by network conditions and link reliability (P_r) in competence and the number of encountered entities by node j (N_j^e) in social contact. The initial estimated trust value at time $t = 0$ is set to 0.5, implying ignorance (complete uncertainty). We report the impact of key design parameters on the four metrics defined in Section 3.4. We vary three key design parameters to examine their effects: (1) the trust threshold (T_{th}); (2) the percentage of compromised nodes (P_c); and (3) the degree of the attack intensity (P_a). In order to model attackers' behaviors in Section 3.1, P_c and P_a are the key design parameters. If a node is selected as compromised based on P_c (i.e., P_c fraction of nodes is compromised), it will perform any attacks described in Section 3.1 with the probability P_a. The scenarios that a malicious node performs attack are explained in detail in Section 3.1.

We allow a 30 min warm-up period (T_w) for peer-to-peer trust evaluation to reach a sufficiently accurate level. We use a 5 min trust update interval (T_u) and a 2 min group communication interval (T_g). For group communications among legitimate member nodes, each node sends out a packet to all nodes whose public keys are available to it. Each data point shown is based on the average of 50 observations of performance data collected during the 4 h of simulation time. All results are shown with 95 % confidence interval (CI) for each data point.

4.2. Schemes for performance comparison

We compare CTPKM with a baseline scheme and two existing schemes. The baseline scheme is a non-trust-based key management which follows all key management procedures in CTPKM except for trust management. The other two existing schemes are selected from the class of

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

Table 3
System parameters and default values.

Parameters	Meaning	Value
GB	Lower bound of the probability that a node behaves well, used in deriving ground truth trust, P_i^X	0.8
α	A weight to consider direct evidence while $(1-\alpha)$ is a weight to consider indirect evidence	0.1
γ	A trust decay factor	0.95
T_{th}	Trust threshold used in the operations of key management and group communication	0.3
TC	Length of a trust chain	4
P_a	Probability that an attacker exhibits malicious behavior, called attack intensity	0.5
T_w	Warm up period in the beginning of network deployment to establish initial trust	30 min
T_u	Trust update interval	5 min
ρ	A constant to normalize the social contact trust	5
LT	The total simulation time	4 h
T_g	Group communication time interval	2 min
P_r	Probability that a link is reliable for transmission	0.99
P_c	Percentage of compromised nodes in a network	20%
R	Wireless radio range	250 m
N	Total number of nodes in a network	92
$1/\lambda$	Average time interval a node joins a network	1 h
$1/\mu$	Average time interval a node leaves a network	4 h
P_d	Standard deviation of a node's average behavior from its actual behavior accounting for behavior variation over time	0.05

certificate-based public key management [20,21], discussed in Section 2.1. The two existing key management protocols are selected for performance comparison against CTPKM for the following reasons:

- The two existing key management schemes fall within the class of certificate-based public key management as CTPKM for fair performance comparison.
- As CTPKM, both schemes use the concept of trust as the basis of decision making such as selecting a trustworthy CA [20] and authenticating the certificate of a public key of a target node based on the web of trust of intermediate nodes in the certificate path [21].
- Chang and Kuo's work [20] represents a centralized public key management with the existence of a trusted party as is often assumed in many existing works. Dahshan and Irvin's work [21] follows the concept of the web of trust as is often used in many existing works to ensure accurate trust assessment of entities in traditional security services such as PGP.

We explain how they are implemented in detail as follows:

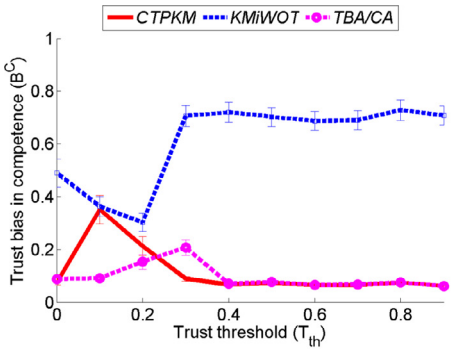
- **Trust-based back-up CA/CA key management (TBA/CA)** [20]: This work uses trust to select the CA and back-up CA (BCA) for key management in MANETs. We tailor it to fit the network environment targeted in this work for fair comparison. In TBA/CA, the most trustworthy node with the highest overall trust value (assuming an equal weight for the three trust components) becomes the CA and the next highest trustworthy node becomes the BCA. When the CA is leaving, the BCA takes the role of the new CA and accordingly a new BCA is selected. Like CTPKM, the trust metric is based on the combination of direct and indirect evidence. However, the indirect evidence used in this scheme is based on the derived measured trust with all nodes (except the target node) serving as the recommenders. This is in contrast to CTPKM which uses only 1-hop neighbors of the target node as the recommenders where indirect evidence is derived based on the recommender's direct experience in order to avoid any impact of compromised nodes on the source of indirect recommendation [10]. Also in TBA/CA, the CA maintains all key pairs and disseminates a key pair to the intended owner regardless of the status of the owner node (whether the node is compromised or not). We apply equal weight to direct and indirect evidence.
- **Key management in web of trust (KMiWoT)** [21]: In this scheme, each node issues a pair of private/public keys and gets a certificate issued by a neighbor node who believes there is a binding between this node's ID and its public key. To compute the trust of a target node, a node first finds a certificate chain of public keys, the last of which is the public key of the target node under trust evaluation. Then the trust value of the target node is computed by the product of trust values of the intermediate nodes on the path of the certificate chain. Essentially trust in KMiWoT means trust in the certificate authenticating a public key belongs to a particular node. A node can obtain the authenticated public key of a target node via two ways: (1) the node itself issues a certificate to the target node who had requested its public key to be certified; (2) the node finds a certificate chain leading to the target node's public key. Trust estimation relies on the existence of a certificate chain to obtain authenticated public keys. If no certificate chain is found, trust will not be updated.

4.3. Comparative performance analysis

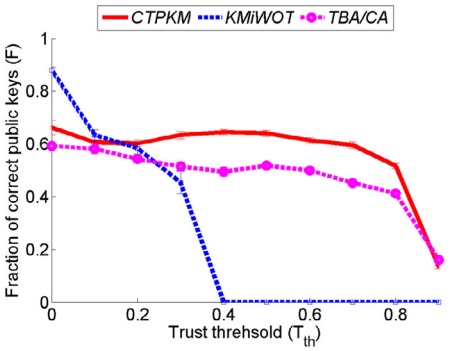
In this section, we compare the performance of CTPKM with non-trust-based and trust-based counterparts including NTB, TBA/CA [20], and KMiWoT [21] under varying parameter values including trust threshold (T_{th}), the percentage of compromised nodes, and attack intensity (P_a).

4.3.1. Effect of trust threshold

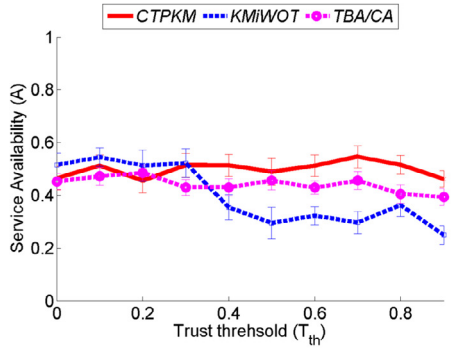
First we examine the impact of a trust threshold T_{th} on the trust bias and four performance metrics as discussed in



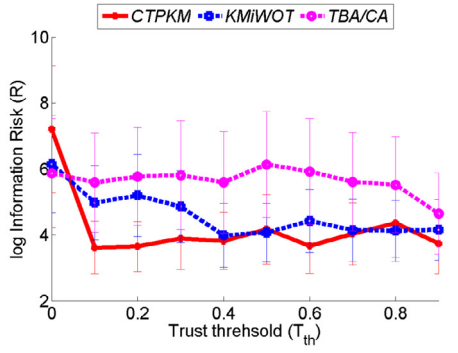
(a) *Trust bias in competence* (B^C) *vs.* T_{th}



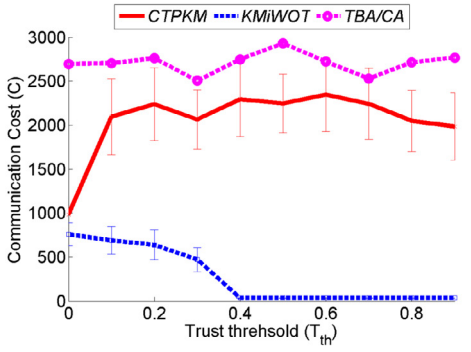
(b) *Fraction of correct public keys* ($\mathcal{F}$) vs. T_{th}



(c) *Service Availability* ($\mathcal{A}$) vs. T_{th}



(d) *Information Risk* ($\mathcal{R}$) vs. T_{th}



(e) *Communication Cost* ($\mathcal{C}$) vs. T_{th}

Fig. 2. Effect of trust threshold (T_{th}) on trust bias and performance.

Section 3.4. In Fig. 2, we compare the performance of three trust-based approaches, CTPKM, KMiWoT, and TBA/CA as all trust-based schemes use a trust threshold as a design feature.

Fig. 2(a) shows trust bias in competence (B^C) of three trust-based schemes. When $T_{th} < 0.4$, B^C is somewhat fluctuating but the overall performance is ordered as CTPKM $\approx$ TBA/CA $>$ KMiWoT. The high inaccuracy of trust estimation in KMiWoT is because peer-to-peer trust estimation is only possible when each node has the certificate of a peer's public key. In KMiWoT, a node can issue a certificate of a peer's public key only when the peer's trust value is higher than a given trust threshold. Thus, if a node's public key certificate is not available, trust cannot be updated. In addition, when two nodes are apart with more than 1-hop distance, the trust of a trustor toward a trustee is calculated based on the sum of trust values obtained from multiple paths (if available) where each trust value is

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

computed based on the product of trust values of all intermediate nodes on the path [21]. Under a highly dynamic environment such as MANETs, a path between two nodes may not exist. In addition, even if there exists any path between two distant nodes, the product of trust values of all intermediate nodes can decay trust values too quickly, which leads to high inaccuracy of trust estimation (i.e., high trust bias) in KMiWoT as shown in Fig. 2(a). The trends of the trust biases in the other two dimensions are similar, thus we do not show them due to space constraint.

In Fig. 2(b), we compare the fraction of correct public keys ($\mathcal{F}$) of the three trust-based key management schemes under varying T_{th}. KMiWoT performs comparably or better than CTPKM and TBA/CA under $T_{th} \leq 0.2$. When $T_{th} > 0.2$, the performance order is as CTPKM > TBA/CA > KMiWoT. In particular, it is noticeable that KMiWoT crashes when $T_{th} > 0.3$ because of high trust bias. The low performance of TBA/CA compared to CTPKM is because TBA/CA does not filter recommendations (i.e., indirect trust evidence) while CTPKM only uses recommendations filtered from trustworthy sources based on T_{th}. In Fig. 2(c), we compare service availability ($\mathcal{A}$) performances of the three schemes. Similar to the trends observed in Fig. 2(b), when $T_{th} > 0.3$, the performance order is observed as CTPKM > TBA/CA > KMiWoT due to high trust bias in KMiWoT and unfiltered recommendations used in TBA/CA.

In Fig. 2(d), we compare the performance of the three schemes in information risk ($\mathcal{R}$). In all cases, lower information risk is observed as T_{th} increases because a high trust threshold only allows public key generation and distribution of highly trustworthy nodes. When $T_{th} > 0.1$, we observe significantly high performance of CTPKM in information security, compared to the other two schemes, with the performance order of CTPKM $\geq$ KMiWoT > TBA/CA. TBA/CA shows the highest information risk because it cannot deal with the case that a CA or back-up CA is compromised. KMiWoT also shows a higher information risk than CTPKM because of high trust bias. In Fig. 2(e), we compare the communication overhead ($\mathcal{C}$) of the three schemes and observed the performance order of KMiWoT > CTPKM > TBA/CA. Although KMiWoT performs the best among three in C, its performances in $\mathcal{F}$ and $\mathcal{A}$ are very low, which offsets the merit of KMiWoT compared to CTPKM.

For the next two sections, we vary the percentage of compromised nodes, P_c, and the degree of attack intensity, P_a, and investigate their impact on the performance metrics. In order to use the same trust threshold in all schemes for fair comparison, we chose 0.3 as the trust threshold even though an optimal trust threshold can be differently selected in each scheme. We set $P_a = 0.5$ and $P_c = 0.2$ unless they are varied for sensitivity analysis.

4.3.2. Effect of percentage of compromised nodes

This section shows the performance comparison of the four schemes as the percentage of compromised nodes, P_c, varies. Fig. 3(a) shows trust bias in competence (B^C) for the three trust-based schemes. In Fig. 3(a), as P_c increases, more compromised nodes exist in the network. In this case, CTPKM performs better than or at least comparable to TBA/CA because more compromised nodes will deter each node from obtaining accurate trust evidence. We observe that KMiWoT performs the worst among three due to the way of trust estimation using excessive trust decay over space and a lack of paths existing between nodes that have each other's public key certificates. In Fig. 3(b), we compare the fraction of correct public keys ($\mathcal{F}$) of the three trust-based schemes and one non-trust based scheme. The performance order is NTB > CTPKM > TBA/CA $\approx$ KMiWoT. Although NTB performs the best among three in this metric, it loses its advantage because it incurs high information risk (see Fig. 3(d)). Compared to the performance of TBA/CA and KMiWoT, CTPKM performs significantly better with the minimum information risk exposed. Fig. 3(c) shows the performance comparison of the four schemes in service availability ($\mathcal{A}$) with the performance order of CTPKM $\geq$ NTB $\approx$ KMiWoT > TBA/CA overall. With the same reason discussed in Fig. 2(d) and (e), the performance order in information risk ($\mathcal{R}$) is as CTPKM > KMiWoT > TBA/CA > NTB in Fig. 3(d) while the performance order in communication cost ($\mathcal{C}$) is KMiWoT > NTB > CTPKM > TBA/CA in Fig. 3(e). KMiWoT incurs the least communication overhead because it leverages the existence of public key certificates for trust estimation which does not introduce extra communication overhead. Also, it does not perform trust assessment in the absence of the public key certificates.

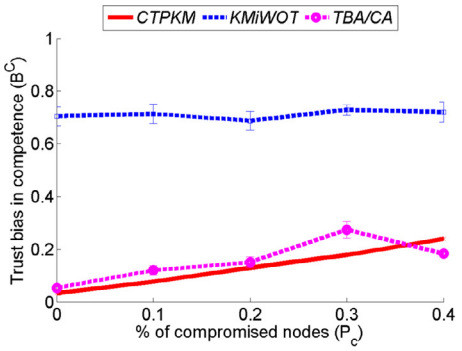
4.3.3. Effect of attack intensity

Lastly this section demonstrates the performance of the four schemes as the attack intensity, P_a, varies. In Fig. 4(a), we observe a similar trend of trust bias in competence of the three trust-based schemes like Fig. 3(a), with the performance order of CTPKM $\geq$ TBA/CA > KMiWoT. In Fig. 4(a), as the attack intensity (P_a) increases, CTPKM significantly outperforms its trust-based counterparts particularly when $P_a > 0.4$. When $P_a \leq 0.4$, TBCA/BA performs better than CTPKM because TBCA/BA uses more trust evidence at the expense of a high communication overhead (since it uses recommendations from all nodes in the network) and can achieve a high accuracy of trust estimation when the attack intensity is low. On the other hand, when the attack intensity is high, $P_a > 0.4$, it helps CTPKM accurately estimate trust because an attacker will consistently exhibit the same bad behavior.

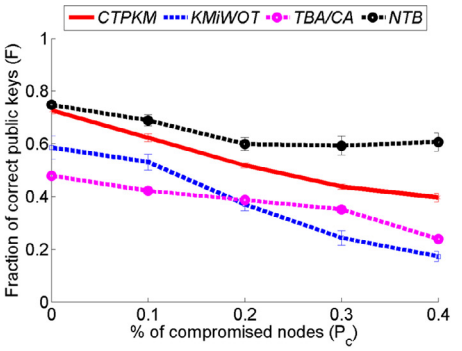
In Fig. 4(b), the trends of performance comparison in $\mathcal{F}$ are similar to those in Fig. 3(b). However, one noticeable difference is that NTB drops its performance with the highest attack intensity, 1, because NTB has no defense feature against high attack intensity. This is also clearly supported by the highest information risk ($\mathcal{R}$) in NTB with the highest attack intensity, as observed in Fig. 4(d).

The performance trends in information risk ($\mathcal{R}$) and communication overhead ($\mathcal{C}$) in Fig. 4(c) and (e) are similar to those in Fig. 3(c) and (e). With DoS attacks, a compromised node can keep requesting public keys of other nodes even if it already has their public keys in order to increase traffic which can waste the network resource. A trust-based key management scheme would make a decision for key management operations such as issuance, distribution, request, and update based on the requesting

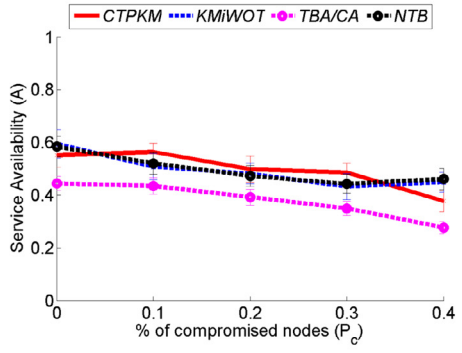
Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014



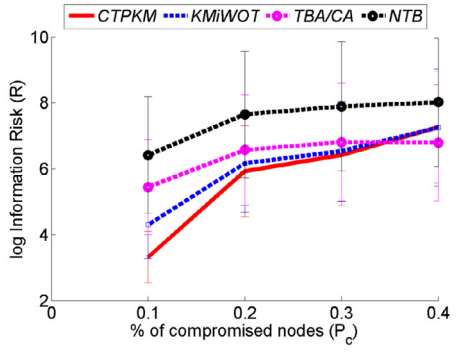
(a) *Trust bias in competence* (B^C) *vs.* P_c



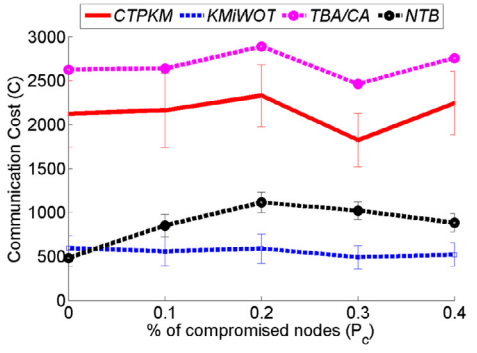
(b) *Fraction of correct public keys* ($\mathcal{F}$) vs. P_c



(c) *Service Availability* ($\mathcal{A}$) vs. P_c



(d) *Information Risk* ($\mathcal{R}$) vs. P_c



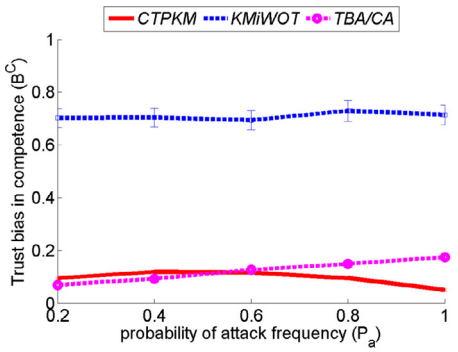
(e) *Communication Cost* ($\mathcal{C}$) vs. P_c

Fig. 3. Effect of percentage of compromised nodes (P_c) on trust bias and performance.

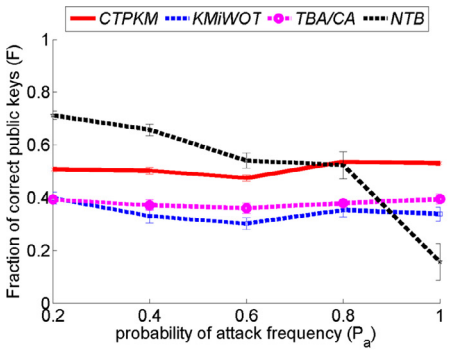
node's trust. Therefore, an accurate trust protocol such as CTPKM can ignore requests issued from "untrustworthy" nodes, thereby effectively thwarting DOS attacks. Consequently, CTPKM is relatively insensitive to the increased percentage of compromised nodes P_c or increased attack intensity P_a in communication cost as shown in Figs. 3(e) and 4(e), respectively. However, a non-trust-based scheme such as NTB will serve all requests even from compromised nodes because it does not use trust for decision making. This causes a significant traffic increase as P_c or P_a increases, as demonstrated in Figs. 3 (e) and 4(e), respectively.

In TBA/CA and KMiWoT, we do not observe much sensitivity of the performance over varying attack intensity. Although KMiWoT has slightly better performance than CTPKM in information risk when the attack intensity is high, the information risk ($\mathcal{R}$) of KMiWoT in general is too high. A reason is that KMiWoT performs poorly in the fraction of correct public keys ($\mathcal{F}$), as shown in Fig. 4(b).

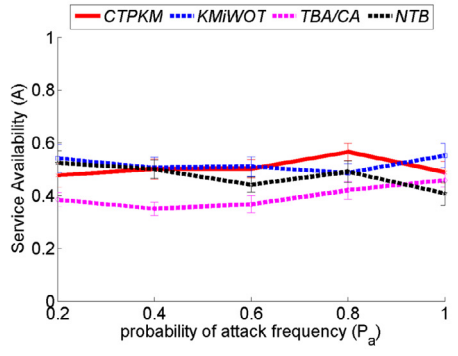
Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014



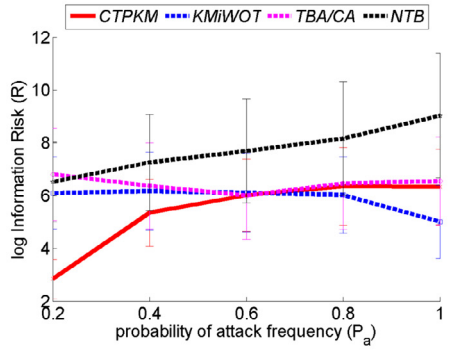
(a) *Trust bias in competence* (B^C) *vs.* P_a



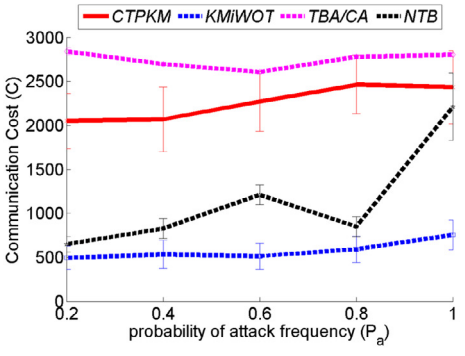
(b) *Fraction of correct public keys* ($\mathcal{F}$) vs. P_a



(c) *Service Availability* ($\mathcal{A}$) vs. P_a



(d) *Information Risk* ($\mathcal{R}$) vs. P_a



(e) *Communication Cost* ($\mathcal{C}$) vs. P_a

Fig. 4. Effect of attack intensity (P_a) on trust bias and performance.

5. Conclusions

In this paper, we proposed a CTPKM for MANETs. Considering three different trust dimensions, namely, competence, integrity, and social contact, CTPKM enables a node to make decisions while interacting with others based on their trust levels. We devised four performance metrics to analyze the impact of our trust threshold based public key management design on security vulnerability (i.e., information risk), availability (i.e., fraction of valid, correct and uncompromised public keys and service availability), and cost (i.e., communication cost).

The design of the proposed trust metric and trust-based key management scheme properly reflects the desirable properties of trustworthy systems for MANET environments [5] discussed in Section 2 in the following way: (1) the use of a trust threshold adjusts potential risk and thus mitigates the impact of risk; (2) trust is measured based on node behavioral evidence in the context of key management; (3) a node decides whether to trust other nodes in the process of key management operations in order to maximize the distribution of valid public keys; (4) a node learns other nodes' trust over time based on past experience in addition to new evidence; and (5) our design

Please cite this article as: J.-H. Cho et al., Trust threshold based public key management in mobile ad hoc networks, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.02.014

enhances both security and performance, leading to a high system reliability.

We conducted a comparative performance analysis of our proposed CTPKM against a counterpart non-trust-based scheme and two existing trust-based key management schemes. We found that CTPKM with a trust threshold design to filter untrustworthy messages or operations can minimize security vulnerability while achieving high availability, without incurring high communication cost. In this work, we assumed a single threshold for node trustworthiness classification. As a future work direction, we plan to investigate more sophisticated fuzzy failure criteria as in [44–46] to further enhance CTPKM performance.

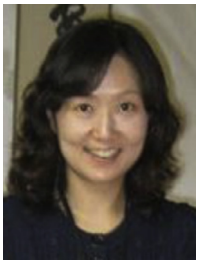
Acknowledgments

This work is supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract no. W911NF-12-1-0445. This research was also partially supported by the Department of Defense (DoD) through the office of the Assistant Secretary of Defense for Research and Engineering (ASD (R&E)). The views and opinions of the author(s) do not reflect those of the DoD or ASD (R&E).

References

[1] K.S. Cook, Trust in society, Russell Sage Foundation Series on Trust, vol. 2, New York, 2003.
[2] M. Blaze, J. Feigenbaum, J. Lacy, Decentralized trust management, in: Proceedings of IEEE Symposium on Security and Privacy, 1996, pp. 164–173.
[3] L. Eschenauer, V.D. Gligor, J. Baras, On trust establishment in mobile ad hoc networks, in: Proceedings of 10th International Security Protocols Workshop, Cambridge, UK, vol. 2845, 2002, pp. 47–66.
[4] J.S. Baras, T. Jiang, Managing trust in self-organized mobile ad hoc networks, in: Proceedings of 12th Annual Network and Distributed System Security Symposium Workshop, San Diego, CA, 2005.
[5] J.-H. Cho, A. Swami, I.-R. Chen, A survey of trust management in mobile ad hoc networks, IEEE Commun. Surv. Tutor. 13 (4) (2011) 562–583.
[6] F. Bao, I.-R. Chen, M. Chang, J.-H. Cho, Trust-based intrusion detection in wireless sensor networks, in: IEEE International Conference on Communication, Kyoto, Japan, 2011, pp. 1–6.
[7] F. Bao, I.-R. Chen, M. Chang, J.-H. Cho, Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, IEEE Trans. Netw. Service Manag. 9 (2) (2012) 161–183.
[8] I.-R. Chen, F. Bao, M. Chang, J.-H. Cho, Trust management for encounter-based routing in delay tolerant networks, in: IEEE Global Communications Conference, Miami, Florida, USA, 2010, pp. 1–6.
[9] R.B. Bobba, L. Eschenauer, V. Gligor, W. Arbaugh, Bootstrapping security associations for routing in mobile ad hoc networks, in: IEEE Global Communications Conference, San Francisco, CA, 2003, pp. 1511–1515.
[10] J.-H. Cho, A. Swami, I.-R. Chen, Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks, J. Netw. Comput. Appl. 35 (3) (2010) 1001–1012.
[11] J.-H. Cho, M. Chang, I.-R. Chen, A. Swami, Trust Management VI, IFIP Advances in Information and Communication Technology, vol. 374, Springer, Berlin Heidelberg, pp. 52–67.
[12] I.-R. Chen, F. Bao, M. Chang, J.-H. Cho, Dynamic trust management for delay tolerant networks and its application to secure routing, IEEE Trans. Parallel Distrib. Syst. 25 (5) (2014) 120–1210.
[13] M. Mahmoud, X. Lin, X. Shen, Secure and reliable routing protocols for heterogeneous multihop wireless networks, IEEE Trans. Parallel Distrib. Syst. 26 (4) (2015) 1140–1153.
[14] E.D. Silva, A.D. Santos, L. Albini, M. Lima, Identity-based key management in mobile ad hoc networks: Techniques and applications, IEEE Wireless Commun. 15 (5) (2008) 46–52.
[15] S. Capkun, L. Buttya, J.-P. Hubaux, Self-organized public-key management for mobile ad hoc networks, IEEE Trans. Mobile Comput. 2 (1) (2003) 52–64.
[16] K.K. Chauhan, S. Tapaswe, A secure key management system in group structured mobile ad hoc networks, in: 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, Beijing, China, 2010, pp. 307–311.
[17] H. Huang, S.F. Wu , An approach to certificate path discovery in mobile ad hoc networks, in: 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.
[18] J. Huang, D. Nicol, A calculus of trust and its application to PKI and identity management, in: ACM 8th Symposium on Identity and Trust on the Internet, Gaithersburg, MD, USA, 2009.
[19] B. Wu, J. Wu, E. Fernandez, S. Magliveras, Secure and efficient key management in mobile ad hoc networks, in: Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium, Denver, CO, 2005.
[20] B.-J. Chang, S.-L. Kuo, Markov chain trust model for trust-value analysis and key management in distributed multicast MANETs, IEEE Trans. Vehic. Technol. 58 (5) (2009) 1846–1863.
[21] H. Dahshan, J. Irvin, A robust self-organized public key management for mobile ad hoc networks, Secur. Commun. Netw. 3 (1) (2010) 16–30.
[22] R. PushpaLakshmi, A. Kumar, R. Rahul, Mobile agent based composite key management scheme for MANET, in: 2011 International Conference on Emerging Trends in Electrical and Computer Technology, Tamil Nadu, India, 2011, pp. 964–969.
[23] N.V. Vinh, M.-K. Kim, H. Jun, N.Q. Tung, Group-based public-key management for self-securing large mobile ad-hoc networks, in: International Forum on Strategic Technology, 2007, pp. 250–253.
[24] L. Xu, X. Wang, J. Shen, Strategy and simulation of trust cluster based key management protocol for ad hoc networks, in: 4th International Conference on Computer Science and Education, Nanning, China, 2009, pp. 269–274.
[25] S. Kent, Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management, Technical Report, Network Working Group, 1993.
[26] A. Shamir, How to share a secret, Commun. ACM 22 (1979) 612–613.
[27] Y.G. Desmedt, Threshold cryptography, Eur. Trans. Telecommun. 5 (4) (1994) 449–458.
[28] L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Netw. Mag. 13 (6) (1999) 24–30.
[29] H. Dahshan, J. Irvine, A trust based threshold cryptography key management for mobile ad hoc networks, in: IEEE 70th Vehicular Technology Conference, Anchorage, AK, USA, 2009, pp. 1–5.
[30] A. Shamir, Identity-based cryptosystems and signature schemes, in: CRYPTO'84, 1984, pp. 47–53.
[31] A. Boudguiga, M. Laurent, Key-escrow resistant ID-based authentication scheme for IEEE 802.11s mesh networks, in: IEEE Wireless Communications and Networking Conference (WCNC'11), 2011, pp. 784–789.
[32] A. Boudguiga, M. Laurent, An EAP ID-based authentication method for wireless networks, in: International Conference for Internet Technology and Secured Transactions (ICITST'11), 2011, pp. 232–239.
[33] Z. Zhao, Z. Zhao, X. Tang, Y. Liu, A new ID-based blind signature from bilinear pairings, in: IET International Conference on Wireless, Mobile and Multimedia Networks, 2006, pp. 1–4.
[34] F.R. Yu, H. Tang, P.C. Mason, F. Wang, A hierarchical identity based key management scheme in tactical mobile ad hoc networks, IEEE Trans. Netw. Service Manag. 7 (4) (2010) 258–267.
[35] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: 9th International Conference on the Theory and Application of Cryptology and Information Security, vol. 2894, 2003, pp. 452–473.
[36] Y. Zhang, W. Liu, W. Lou, Y. Fang, Securing mobile ad hoc networks with certificateless public keys, IEEE Trans. Depend. Secure Comput. 3 (4) (2006) 386–399.
[37] A.M. Arokiaraj, A. Shanmugam, ACS: An efficient address based cryptography scheme for mobile ad hoc networks security, in: International Conference on Computer and Communication Engineering, 2008, pp. 52–56.
[38] Y. Zhang, W. Liu, W. Lou, Y. Fang, Y. Kwon, AC-PKI: anonymous and certificateless public-key infrastructure for mobile ad hoc networks, in: IEEE International Conference on Communications (ICC'05), vol. 5, 2005, pp. 3515–3519.
[39] H. Sun, X. Zheng, Z. Deng, An identity-based and threshold key management scheme for ad hoc networks, in: International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, Hubei, China, 2009, pp. 520–523.

[40] L.-C. Li, R.-S. Liu, Securing cluster-based ad hoc networks with distributed authorities, IEEE Trans. Wireless Commun. 9 (10) (2010) 3072–3081.
[41] Dartmouth University, Mobility traces data from crawdad (a community resource for archiving wireless data at Dartmouth).
[42] M.H. MacDougall, Simulating Computer Systems, Computer Systems Series, The MIT Press, 1987.
[43] J.-H. Cho, I.-R. Chen, P. Feng, Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad hoc networks, IEEE Trans. Reliab. 59 (1) (2010) 231–241.
[44] I.-R. Chen, F. Bastani, Effect of artificial-intelligence planning-procedures on system reliability, IEEE Trans. Reliab. 40 (3) (1991) 364–369.
[45] F.B. Bastani, I.-R. Chen, T. Tsao, Reliability of systems with fuzzy-failure criterion, in: Annual Reliability and Maintainability Symposium, Anaheim, CA, 1994, pp. 442–448.
[46] I.-R. Chen, F. Bastani, T. Tsao, On the reliability of AI planning software in real-time applications, IEEE Trans. Knowl. Data Eng. 7 (1) (1995) 4–13.



Jin-Hee Cho received the B.A. degree from the Ewha Womans University, Seoul, Korea and the M.S. and Ph.D. degrees in computer science from the Virginia Tech. Since 2009, she has been working as a computer scientist at the U.S. Army Research Laboratory, Adelphi, MD. Her research interests include wireless mobile networks, mobile ad hoc networks, sensor networks, secure group communications, group key management, network security, intrusion detection, performance analysis, trust management, cognitive networks, social networks, dynamic networks, and resource allocation. She received the best paper awards in IEEE TrustCom09 and BRIMS13. She received the 2015 IEEE Communications Society William R. Bennett Prize in the Field of Communications Networking. She is selected to receive the Presidential Early Career Award for Scientists and Engineers (PECASE) in 2016. She is a senior member of the IEEE and a member of the ACM.

Ing-Ray Chen received the B.S. degree from the National Taiwan University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in computer science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, wireless systems, security, trust management, intrusion detection, and reliability and performance analysis. He currently serves as an editor for IEEE Communications Letters, IEEE Transactions on Network and Service Management, The Computer Journal, and Security and Network Communications. He is a recipient of the 2015 IEEE Communications Society William R. Bennett Prize in the field of Communications Networking. He is a member of the IEEE and ACM.

Kevin S. Chan received the B.S. degree in electrical and computer engineering (ECE)/EPP from Carnegie Mellon University (Pittsburgh, PA) and the Ph.D. degree in ECE and MSECE from Georgia Institute of Technology (Atlanta, GA). He is a research scientist with the Computational and Information Sciences Directorate at the U.S. Army Research Laboratory (Adelphi, MD). Previously, he was an ORAU postdoctoral research fellow at ARL. His research interests are in network science, with past work in dynamic networks, trust and distributed decision making and quality of information through ARL's Network Science Collaborative Technology Alliance and Network and Information Sciences International Technology Alliance.